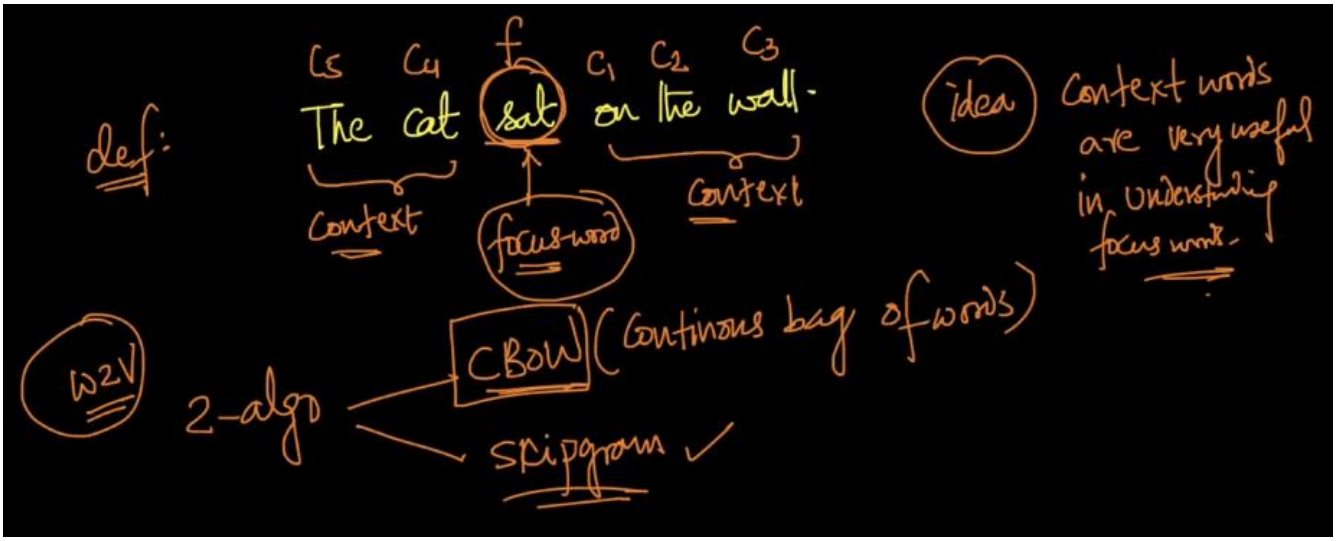
Word2Vec

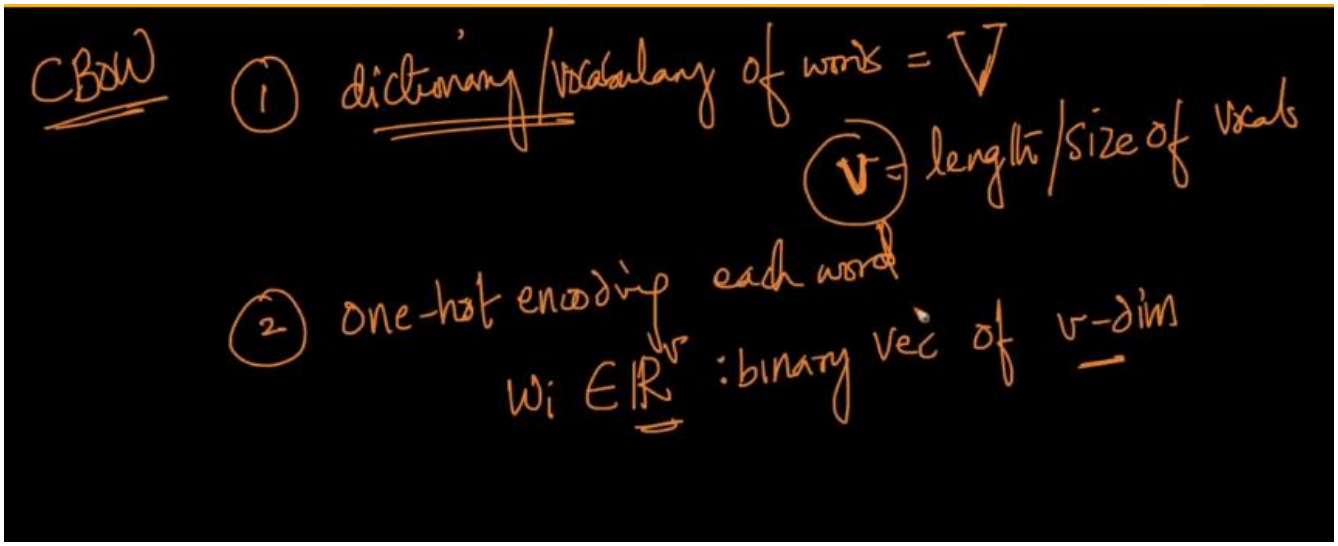
Word2Vec is not a deep learning algorithm. Here we determine the focused word and context words(Words surrounding the focus word).

Core Idea: Context words are very useful in understanding the focus words.
Two Word2Vec Algorithms.



CBOW: Given context words can we predict the focus word.

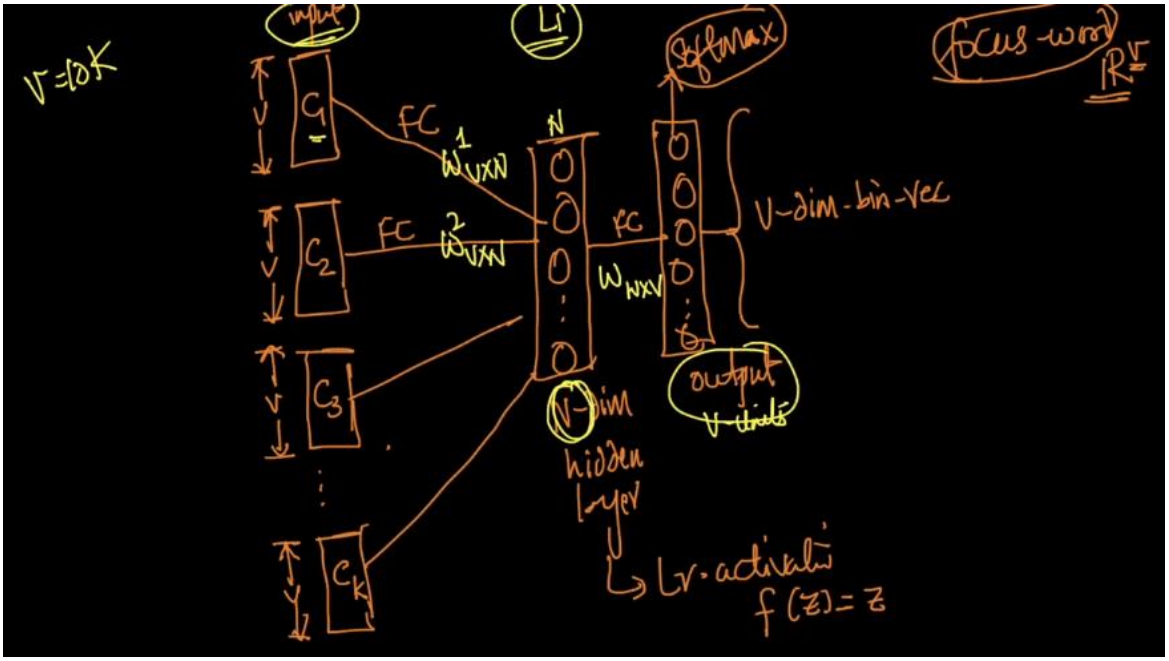
Step1:
Dictionary/Vocabulary of words.



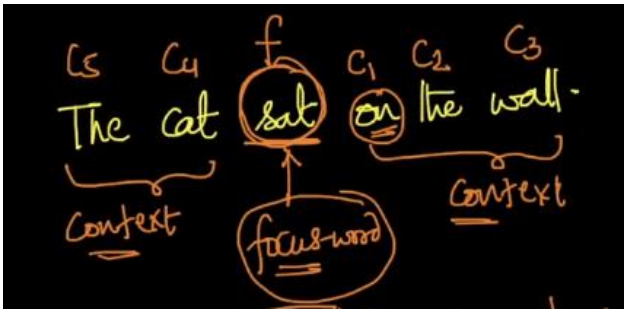
Each of the word is a “v” dimensional one hot vector, and we add the hidden layer of “N” dimensional vector.
Here, the weight is “V*N”, between the input layer and first hidden layer.

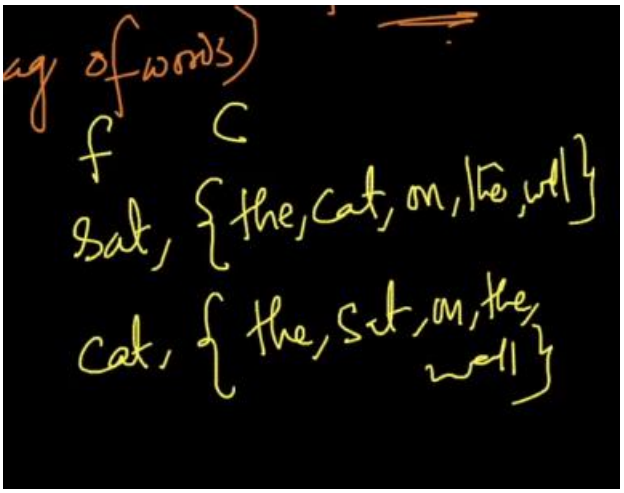
The output layer is of “V” dimensional because, we are trying to predict the “V” dimensional **focus vector by using the soft-max layer to predict the word.**

The output layer is binary output, as the focus word is determined by the softmax function.



The way you train the CBOW is as follows:
We will create the Training data by generating the pairs of **context words** and **focus words.**

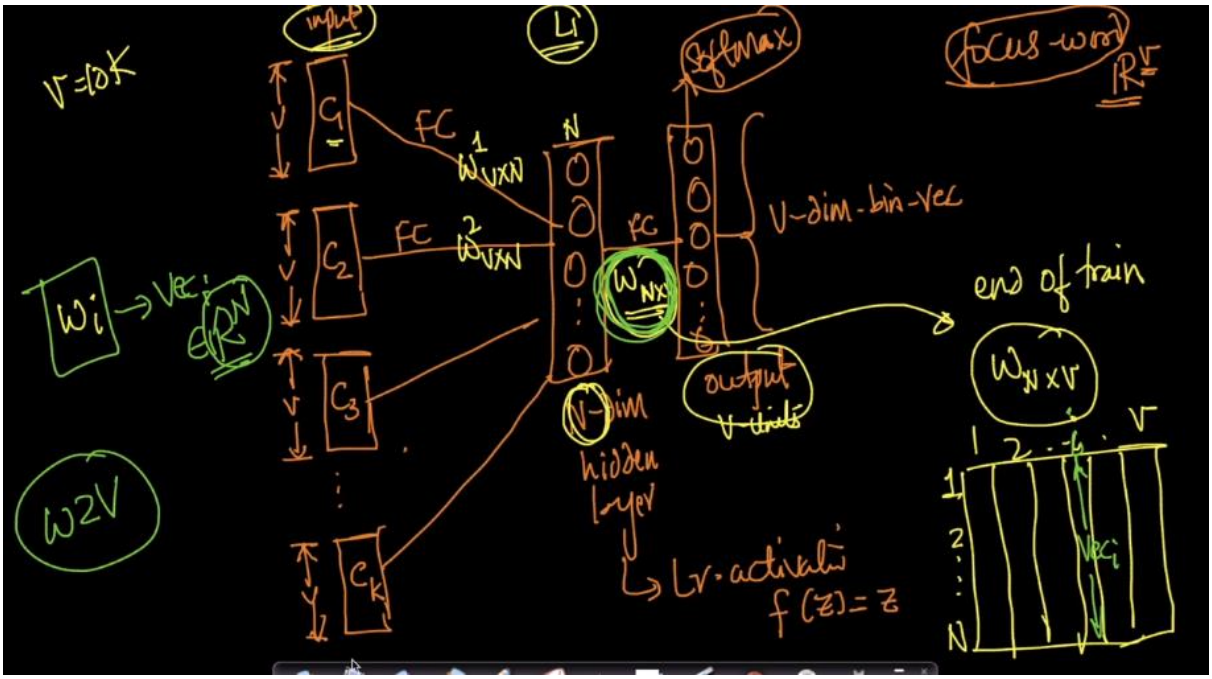




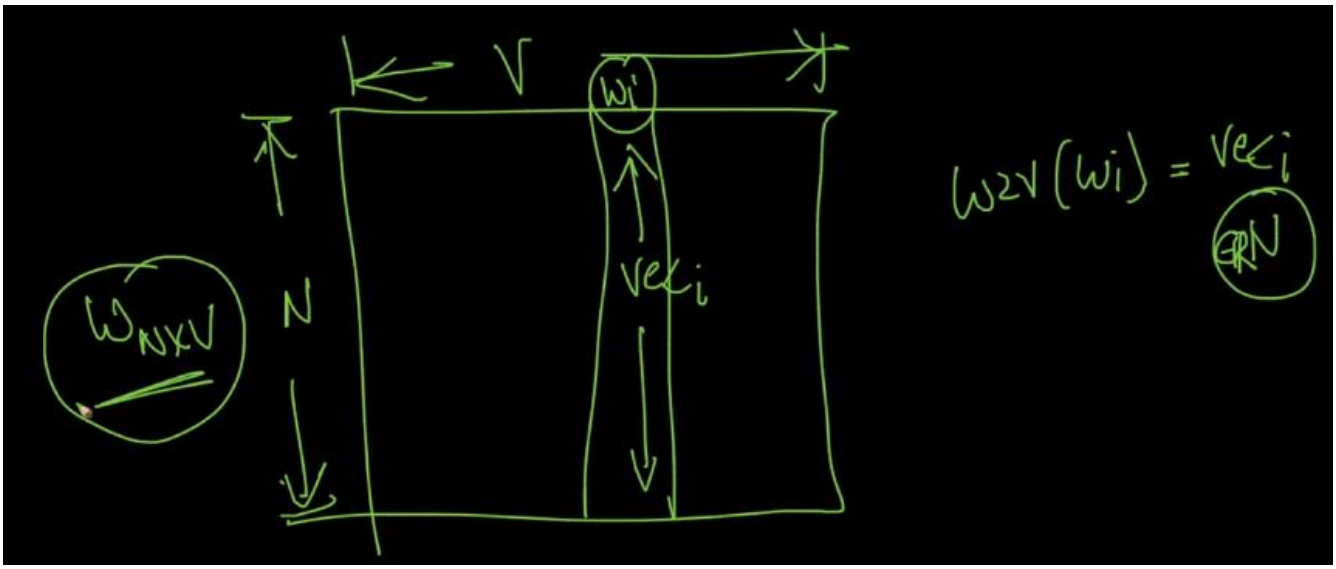
We will train this CBOW words, at the end of this training we will have all of the weights.

The weights just before the output layer is considered as the representation of the “N” dimensional vector.

Therefore, we get the “N” dimensional representation of the each word.



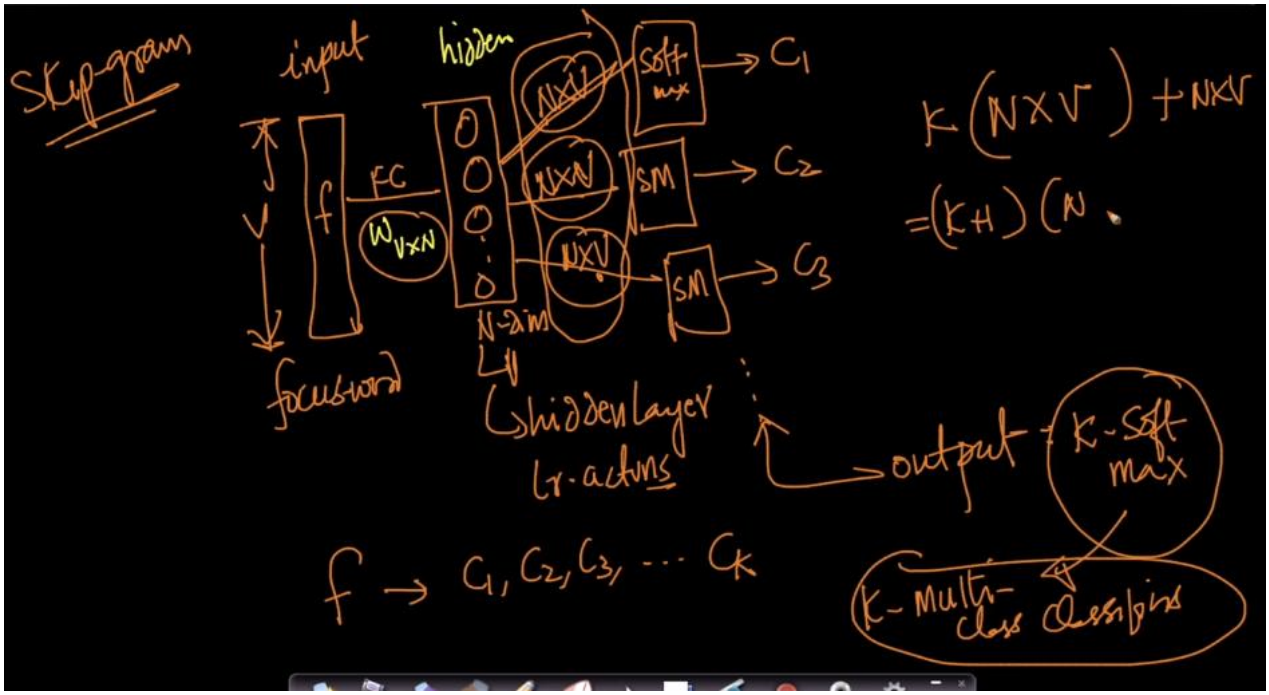
Matrix representation of the weights in detail.



Word2Vec: Skip-gram

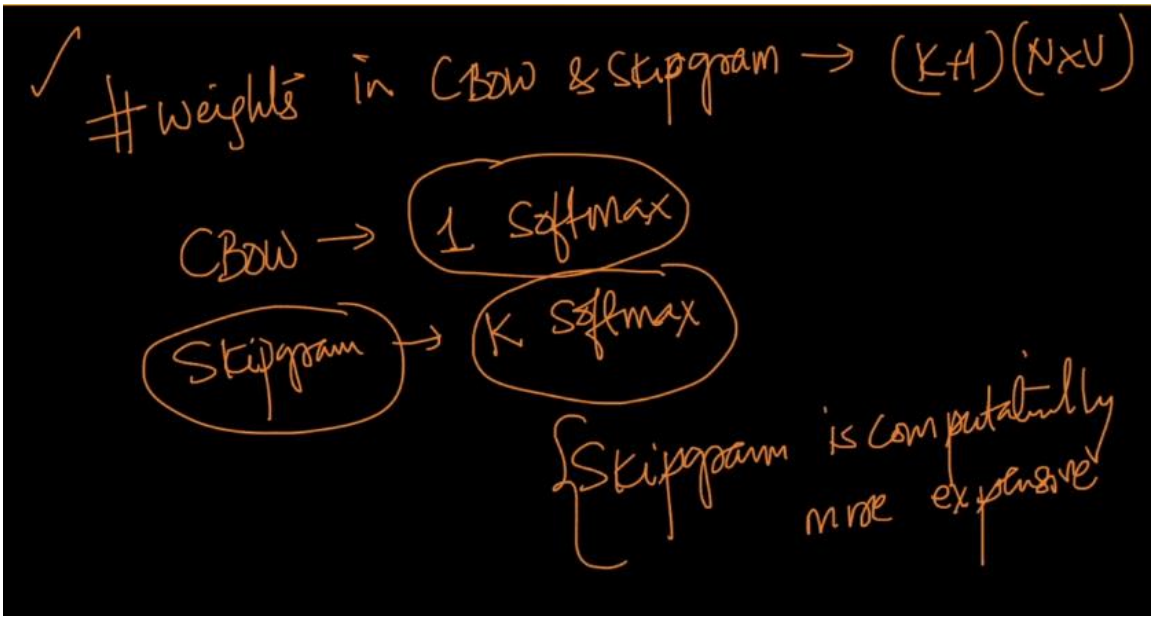
We will try to predict the context words, given focus word. This looks exactly as the flipped CBOW.

Here we get the K softmax outputs, with C1, C2 and so on.. This is like the K-Multi class classification.



Here we have the number of weights in CBOW and Skip-gram → (K+1)*(N*V).

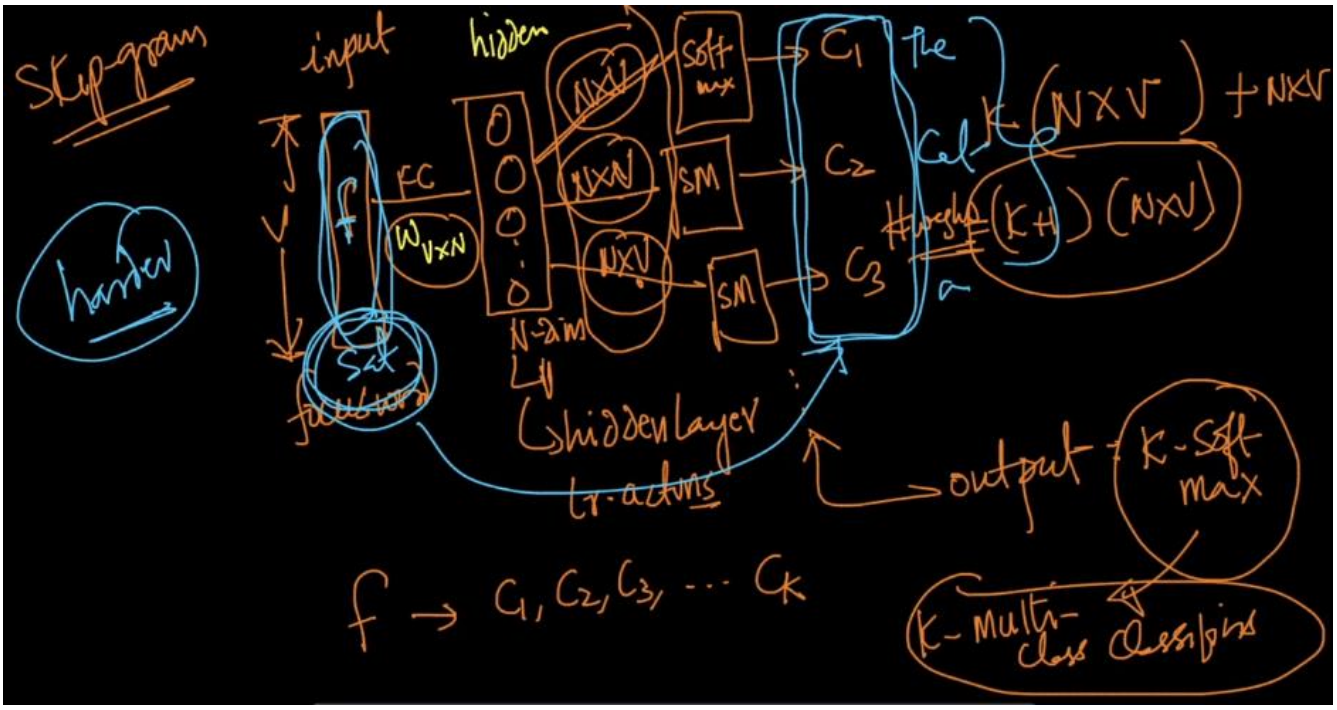
Skip gram takes more time, this is computationally more expensive.



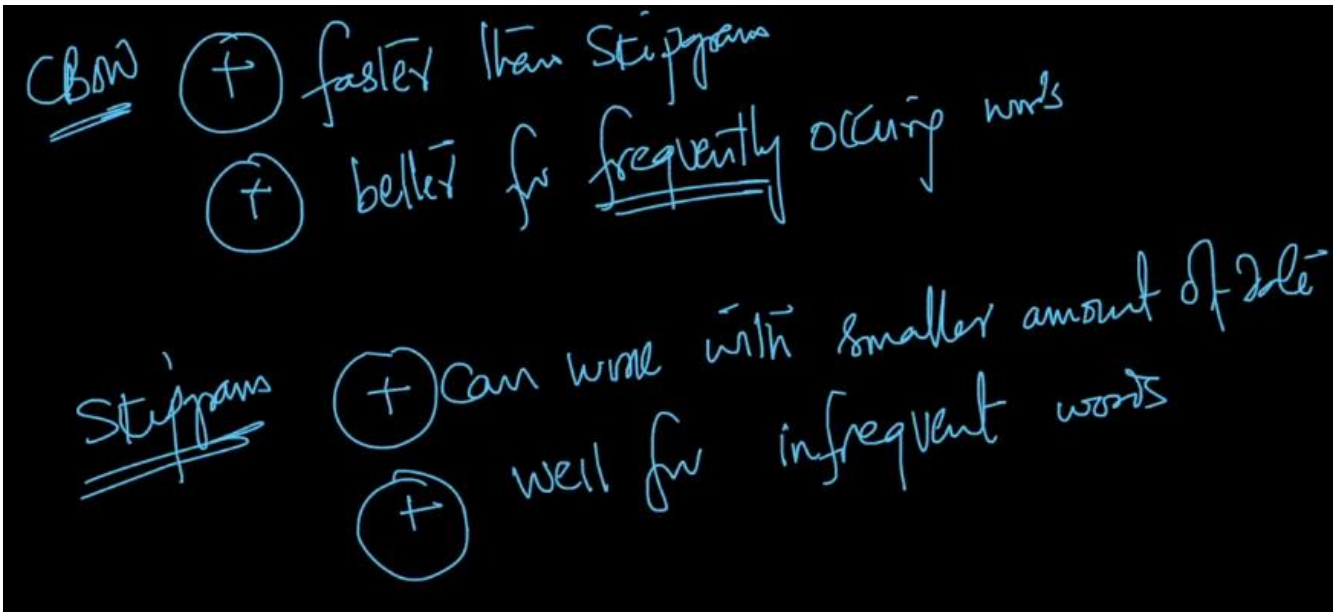
Comparison between the CBOW and Skip-gram:

CBOW is more like fill in the blanks.
Skip-gram is more harder as we want to construct the sentence based on the focus word.

Though Skip-gram is more harder problem to solve, It is more powerful.

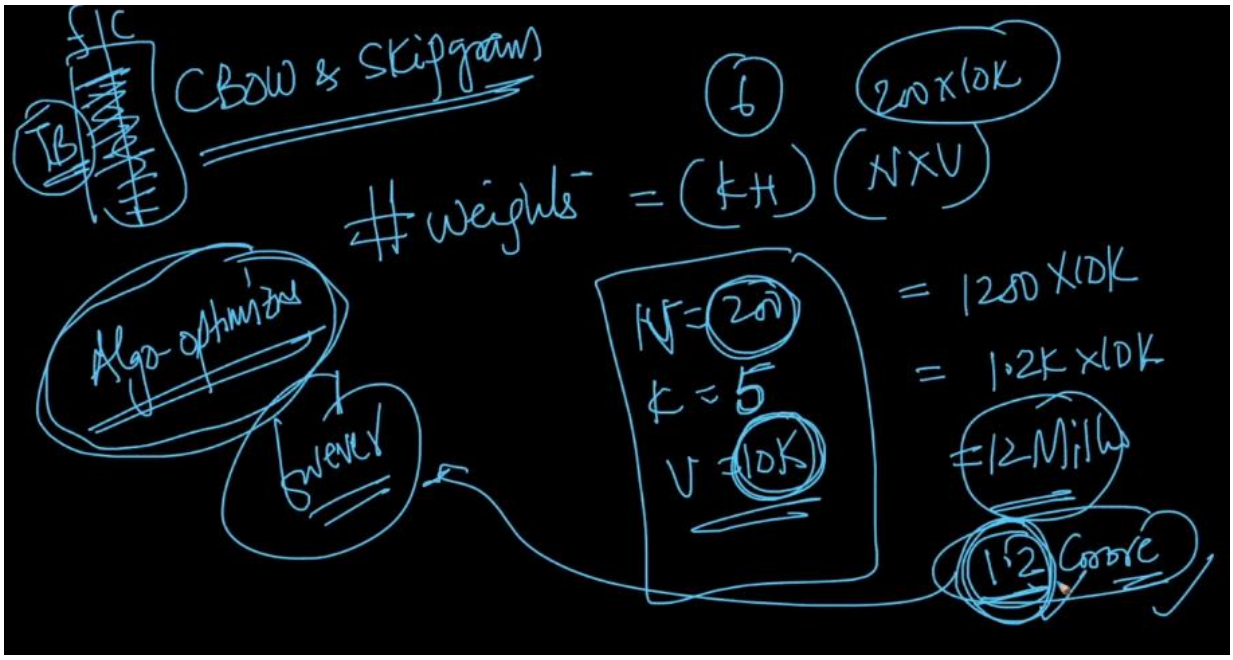


For both skip gram and CBOW, as K increases the more the context words.



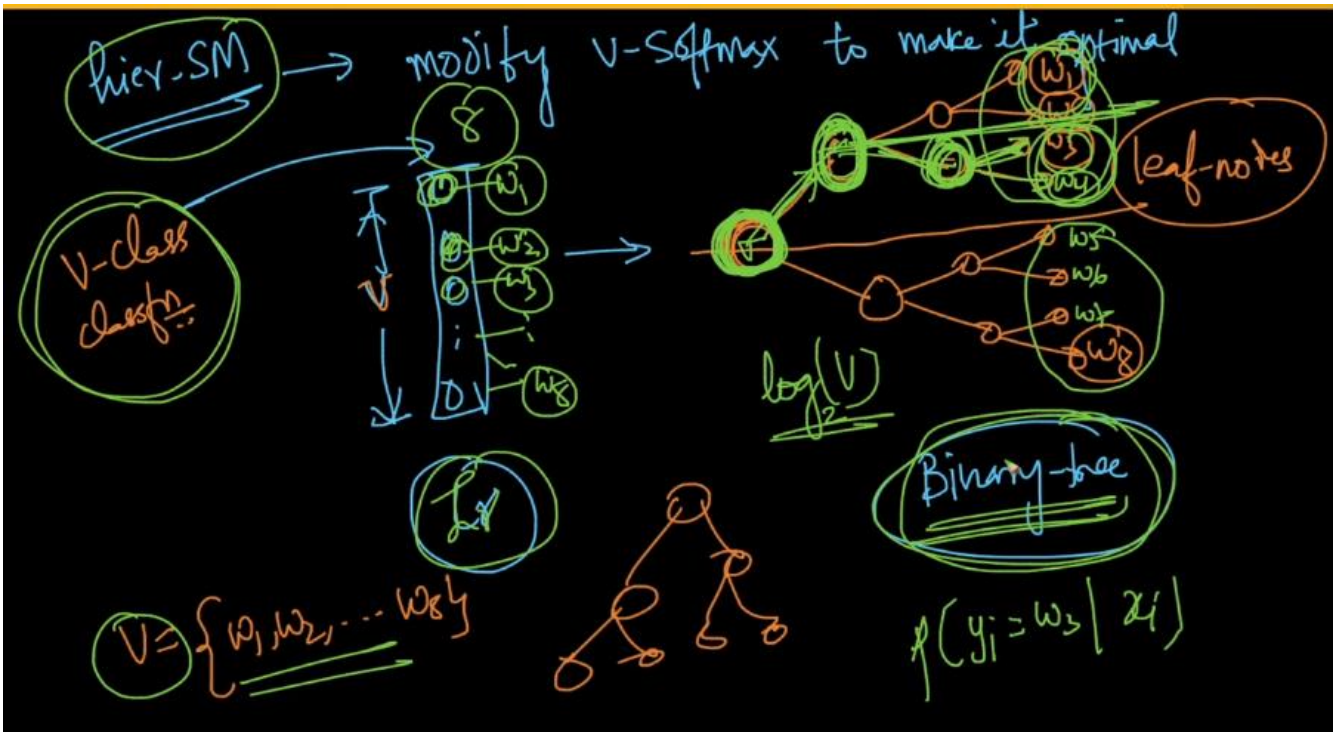
With both these we have the problem.

Here we have to train more number of weights in both the cases for a smaller dataset.



Algorithmic optimizations: Here we have the two optimization techniques Hierarchical softmax and negative Sampling.
Can we optimize the V dimensional Softmax?

First Idea:



Second Idea:

We update only a sample of words, per iteration.

1. We define the sample by always keep the target word.
2. Non – target words, we can sample problematically.